

Localisation of Drone Controllers from RF Signals using a Deep Learning Approach.

David Shorten
University of Hull
Hull, United Kingdom
David.Shorten@hull.ac.uk

Saket Srivastava
University of Lincoln
Lincoln, United Kingdom
ssrivastava@lincoln.ac.uk

John Murray
University of Hull
Hull, United Kingdom
john@jcmurray.com

ABSTRACT

Despite their many uses, small commercial Unmanned Aerial Systems (UASs) or drones pose significant security risks. There is, therefore, a need to find methods of detecting, localising and countering these vehicles. This paper presents work towards autonomously localising drone controllers from the Radio Frequency (RF) signals they emit. An RF sensor array is used to monitor the signal spectrum. A Convolutional Neural Network (CNN) is trained to be able to predict the bearing of the drone controller, relative to the sensor, given its output. The position of the controllers can then be calculated from these bearings, provided that at least two such sensors are deployed a reasonable distance apart. The model is able to achieve a mean absolute error of 3.67° in bearing calculation, which translates into a moderate positional error of 40m at a range of 500m.

CCS CONCEPTS

• **Computing Methodologies** → **Machine learning**; • **Computer systems organization** → *Embedded and cyber-physical systems*;

KEYWORDS

Drone, UAS, RF, Radio, Surveillance, Security, Deep Learning, Convolutional Neural Network

ACM Reference Format:

David Shorten, Saket Srivastava, and John Murray. 2018. Localisation of Drone Controllers from RF Signals using a Deep Learning Approach.. In *Proceedings of 2018 International Conference on Pattern Recognition and Artificial Intelligence (PRAI'18)*. ACM, New York, NY, USA, Article 4, 8 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Despite their popular image as toys, commercial Unmanned Aerial Systems (UASs or Drones) are increasingly being used in dangerous and nefarious activities. There have been reported incidents of drones being used to smuggle contraband into prisons in the USA, Brazil, Ireland, Greece, Russia, Switzerland, Australia and the UK [4, 8]. Drones are particularly useful in this setting as the smuggler

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PRAI'18, August 2018, New Jersey, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

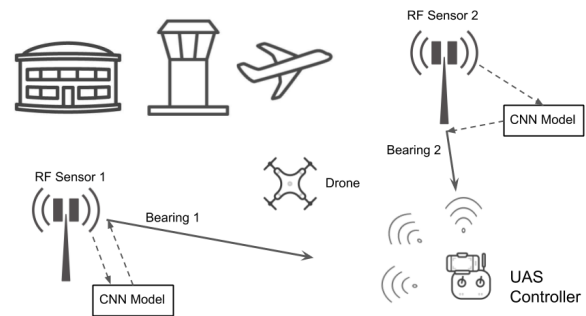


Figure 1: Schematic diagram of our system. Two RF sensors receive the signals from the UAS controller. This data is fed into each sensor's copy of the CNN model, which calculates the bearing of the controller.

can deliver the contraband from a safe distance, whilst receiving a live video feed from the drone, allowing them to respond to conditions in the vicinity of the prison. In the UK, a single gang was recently convicted of flying contraband worth £1 million into prisons in 49 separate flights [4]. One prison governor in the UK has described drones as a “gamechanger” that “came from nowhere” [4]. When drones are flown in the vicinity of airports, the consequences can be severe. Here they pose a significant collision risk to the larger aircraft. The scope of this problem is substantial. In the three months from July to September 2017 alone, over 500 incidents of “Hazardous and/or Unauthorized UAS activity” were reported by pilots to the US Federal Aviation Authority [2]. In the United Kingdom, the Airprox Board determined that, in 2016, there were 65 “risk-bearing” incidents between UASs and manned aircraft [7]. A threat which is yet to be fully realised is the use of drones in terrorism and other forms of unconventional warfare. ISIS have used drones to drop explosives on civilian and military targets in Syria [34]. Official warnings as to the potential for the use of drones in terrorist activities have been issued by many sources, including a British Prime Minister [31].

Despite these risks, there exists a large number of instances in which drones can have a dramatic, positive impact on human life. Examples of these include the automated inspection of building sites [28], data collection for precision agriculture [20] and search and rescue [5]. Given these significant beneficial uses for commercial UASs, a ban on the technology would be harmful. It is, therefore, of paramount importance that technologies are developed which can mitigate the risks of drones, whilst allowing their legitimate and beneficial use to thrive. Due to this pressing need for effective UAS risk mitigation techniques, significant work has already been

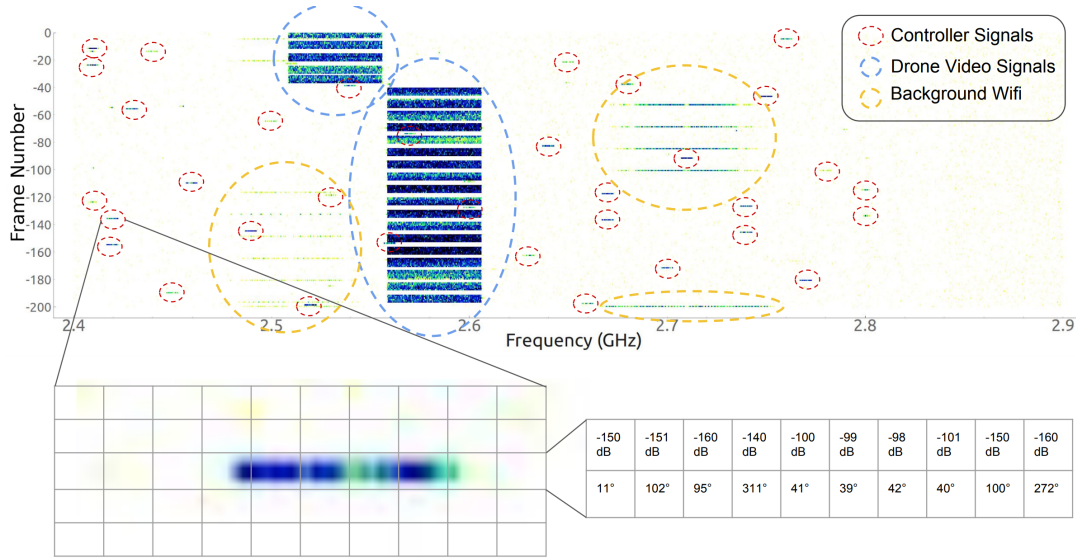


Figure 2: Waterfall diagrams of the RF power spectrum while a DJI Inspire 1 was flying. Time is represented on the vertical axis and frequency on the horizontal. Colour represents the RF power at that point in time and frequency, white being the lowest and blue the highest. We have also included a diagram of the discretisation process of the signal into frequency bin, at a given point in time. The bins in which the control signals reside can be found from their higher power, and their corresponding angles extracted.

conducted towards their development. We identify two principle components to the mitigation of risks from UASs. The first of these is the detection of the presence of the UAS together with the localisation of the vehicle and controller. The second is interdiction methods which are required in order to destroy, capture, or hinder the normal operation of the drone. This work focuses on a method for the localisation of UAS controllers. It is worth mentioning that, in some contexts, the interdiction of the drone is not of paramount importance. For example, in the case of prison contraband smuggling, whilst the primary focus is to stop contraband entering the prisons, the secondary goal is to apprehend the criminal gangs who are orchestrating these smuggling crimes. Similarly, in the context of airports, it is imperative that the operators of drones can be caught and punished in order for an effective deterrence to be established. In order to successfully apprehend drone operators, the most important capability within a counter-UAS system is the rapid and accurate localisation of the drone controllers. Therefore the primary focus of this work is to develop a deep learning system that can autonomously predict the location of a wireless drone operator near sensitive installations such as airports and prisons.

A number of different sensor types are available for the detection and localisation of UASs and their controllers. The sensor types that have received substantial attention from the research community and commercial providers are: Radio Frequency (RF) [26, 30], Electro-Optical (EO) [13], acoustic [23] and radar [25]. As this work focuses on the localisation of drone controllers, RF sensors must be used. The commercial counter-UAS systems of which the authors are aware require a human-in-the-loop in order to monitor the output of the sensors and make decisions on the presence and location of drones and their controllers. There are a number of problems

with this approach. Firstly, sites such as prisons and airports require 24-hour protection. Employing the number of skilled operators required in order to meet this requirement is costly. Furthermore, the performance of human operators monitoring the output of sensors is known to be low, due to limitations in human attention span [19]. Finally, it is not guaranteed that the human operator will be able to interpret the output of the sensors optimally. For example, neural networks can outperform cardiologists in the interpretation of data from electrocardiograms [27].

In the context of the current problem (locating drone controllers), the authors' industrial partners in this project (discussed further in Section 3) have already attempted to solve the challenge using traditional signal processing techniques, with limited success. Therefore, it is imperative that in order to avoid the human-in-the-loop element and make the system completely autonomous, we implement a machine learning technique for computationally performing localisation. Dependence on human input would be a great disadvantage for this system, due to the prohibitive cost of manning it around the clock. Machine learning has proven extremely successful in a number of domains in which hand-coded solutions have failed and where large amounts of labelled data can be gathered [15]. In this task, data labels can be easily generated by co-locating a GPS receiver with the controller, as specified in Section 3.3. Moreover, once a successful model has been found, it can easily be adapted to new requirements (eg: new drone controller types), by training it on an updated training set.

To the best of the authors' knowledge, there has only been a single study applying machine learning to RF data for the purpose of detecting drones [30]. That being said, machine learning has been applied to electro-optical [13, 29], acoustic [6, 21, 23], and



Figure 3: Picture of RF sensor (Hyperion) hardware in deployment.

radar [14, 16, 25] data for this purpose. A mixture of techniques have been used in that work, namely: K-nearest-neighbours [23], support vector machines [6, 14], adaptive Bayesian methods [16] and deep learning [13, 21, 25, 29, 30]. All this work has focused on the detection of UASs, as opposed to the localisation of the drones or controllers. In much of this work [6, 14, 16, 23, 25], the training and/or test data was not collected in real-world situations. The remaining work [13, 21, 29, 30] suffered from low accuracy and/or precision. The work presented in the remainder of this paper focuses on the localisation of drone and controller, which is absent in this current literature. This is important in the future of UAS regulation, prevention and control especially with regard to the successful prosecution of offenders.

The eventual goal of this research is to produce models which are capable of allowing the sensors to operate fully autonomously. That is, we aim to produce models which can predict the presence of drones as well as the bearing of detected drones and controllers, given the output of a single sensor. If the models are running on two spatially separated sensors, then the locations of the controller and drone can be calculated from these bearing values.

This paper reports on our work towards predicting the bearing of the controllers. Work on detection as well as the bearing of drones is ongoing and will be reported on in future publications.

2 UAS RF SIGNALS

UASs use radio signals for the transmission of the operator's control instructions as well as for sending video and telemetry data from the drone back to the controller. Most widely available small UASs operate in the license-exempt bands of 2.4000 – 2.4835GHz and 5.470 – 5.725GHz [33]. In this work, we focus on the localisation of drone controllers operating in the 2.4000 – 2.4835GHz

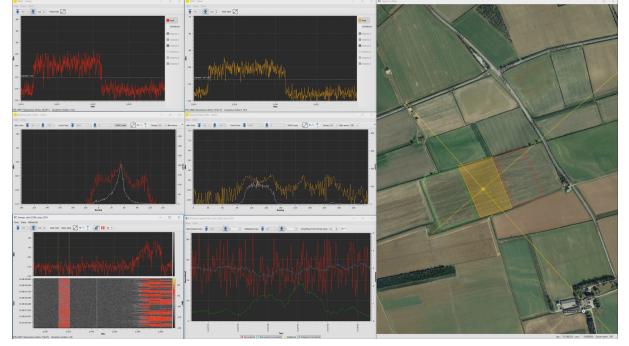


Figure 4: Screen-shot of the control and visualisation software of the Hyperion signal detector produced by Metis Aerospace (metisaaerospace.com).

band, as this is where the control signals of the majority of popular commercial drones are found. However, our approach can be easily extended to operate in other bands. Figure 2 shows a visualisation of measurements of drone video and control signals using our hardware. In this figure, one can clearly see the Frequency-Hopping-Spread-Spectrum (FHSS) technique [24] employed by the controller, whereby the frequency used is constantly changing within the band. It is interesting to observe that, at least in this instance, the controller signals are timed to occur when the video signal is not being transmitted. We have observed this pattern in all the drones which we have studied so far. The authors believe that it is likely that this feature is used by the trained neural networks in order to isolate the controller signals.

3 METHODOLOGY

This work was conducted with the lead project partners Metis Aerospace¹, based in Lincoln, UK. Metis have developed an RF sensor product called Hyperion, capable of detecting RF signals from drones and their controllers. This sensor is able to calculate the RF signal power and bearing within frequency bins. A picture of the sensor in deployment can be found in Figure 3 and a screenshot of the associated software suite may be found in Figure 4. The system is currently being deployed in the UK with Person-In-The-Loop (PITL). At present, using scripts based on signal-processing techniques, along with operator input, they are able to detect the presence of drones within a roughly 1km radius of the sensor. They are also able to isolate the location in time and frequency in which the signals of the drones and controllers reside. They can then use the bearing values within those bins to give the bearings of the controller and the drone, relative to the sensor. By using two sensors placed a reasonable distance apart, and by calculating the bearing of a drone or controller relative to each sensor, they can then calculate the position of the drone or controller. This is shown Figure 1.

¹<http://metisaaerospace.com/contact-us/>
Boole Technology Centre, Beevor Street, Lincoln, LN6 7DJ, United Kingdom

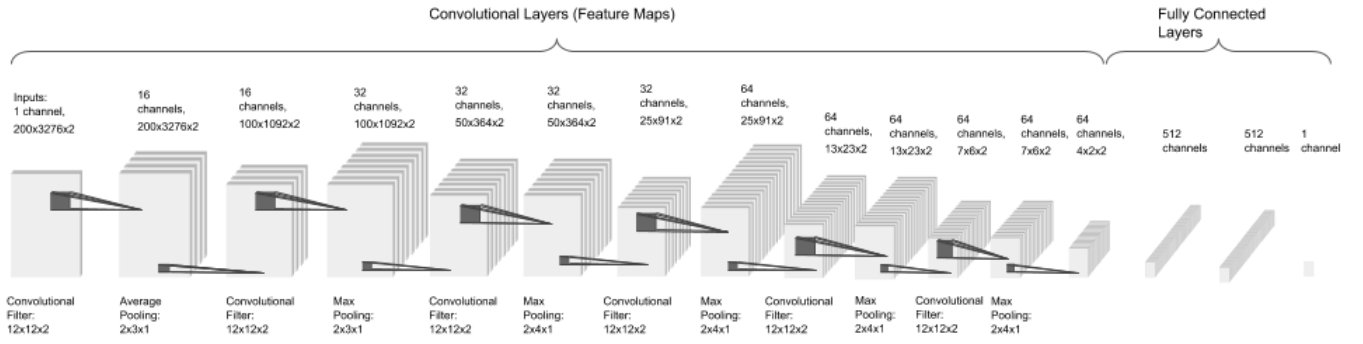
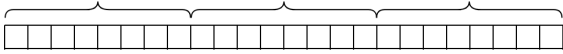


Figure 5: Diagram of the CNN architecture. 3D blocks represent the inputs and these are stacked into multiple channels as features are extracted. Notice the decrease in the dimension of the inputs and the increase in the number of channels across the network.

Simple Window Construction



Augmented Window Construction

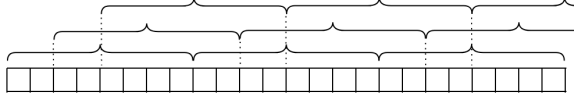


Figure 6: Diagram showing our method of data augmentation by time window construction. The sequence of frames needs to be divided into windows of fixed length in order to be fed into the CNN. Instead of dividing the sequence of frames into non-overlapping windows (top) we divide it into windows with a substantial amount of overlap (bottom). When the model is deployed, it uses the un-augmented method of window construction (top).

3.1 RF Sensor Hardware

The Hyperion sensor (shown in Figure 3) consists of an omnidirectional antenna as well as a direction-finding antenna array. This work makes use of the direction-finding array, which consists of six circularly polarised antennae arranged hexagonally. The antennae are each connected to a six-way RF switch, which provides a common feed into the system receiver, a CRFS Nexus R-8. Under control of the receiver, the RF switch selects each of the six antennae in turn. The RF signal is routed into the receiver where the signal is digitised and the incident angle calculations are carried out.

For this work, the sensor is configured so as to produce readings in bins 30.5176kHz wide, between the frequencies 2.4GHz and 2.4998GHz. This resulted in a total of 3276 bins. For each bin, two readings are produced: the power and the bearing of the available signal in that bin. The sensor operates at a rate of around 300 frames per second, where one frame is the power and bearing readings for all 3276 bins.

3.2 Model

The hardware setup described in Section 3.1 produces around 7 billion floating point values (bearings and amplitudes) per hour. Moreover, data augmentation techniques described in Section 3.5, are easily capable of increasing this by 2 orders of magnitude. Furthermore, the dimension of the inputs is very large. A given model would be required to accept inputs of dimension $(t, 3276, 2)$, where t is the number of data frames being fed into the model. Deep learning has been demonstrated to offer the best performance on high dimensional tasks with large training sets [9, 15] and, as such, it is most appropriate machine learning technique to use in this work.

We decided to use a Convolutional Neural Network (CNN) as opposed to a Recurrent Neural Network (RNN). Traditionally, RNNs have been a popular choice when applying neural networks to the time series domain [11, 12]. This is partly due to the fact that, in theory at least, they are able to remember events that occurred infinitely far back in time. However, recent work has demonstrated that CNNs can achieve better performance on a majority of sequence modelling [3] and video classification benchmarks [35]. They have also been responsible for recent ground-breaking results on time-series modelling [32].

In our preliminary work, we optimised architectures for both CNNs and RNNs. Work on RNNs involved LSTM [18] layers. We experimented with both traditional convolutional layers, operating over individual frames, connected to LSTM layers, along with convolutional LSTMs [36], where the convolutional layers were themselves LSTMs. We found that the CNN architectures which we optimised were able to achieve substantially lower error, whilst also training faster.

The specification of the model architecture can be found in Table 1 along with a diagram in Figure 5. The associated hyperparameters can be found in Table 2. Here, you will notice that we have followed the advice of [3] in using dilated convolutions, in order to handle the very large size of the input which the model received. Another interesting feature of the model is the use of average pooling in the first layer. Our motivation for experimenting with average pooling was that radio data is very noisy. This is particularly severe in the part of the spectrum in which our model operates (2.4 - 2.4998GHz),

Layer Name	Layer Type	Filter Dimensions	Di-	Stride Size	Dilation Size	Output Channels	Output Dimensions
Convolution 1	Convolution	12×12×2		1×1×1	3×3×1	16	200×3276×2×16
Pooling 1	Average Pooling	2×3×1		2×3×1	1×1×1	16	100×1092×2×16
Convolution 2	Convolution	12×12×2		1×1×1	3×3×1	32	100×1092×2×32
Pooling 2	Max Pooling	2×3×1		2×3×1	1×1×1	32	50×364×2×32
Convolution 3	Convolution	12×12×2		1×1×1	2×2×1	32	50×364×2×32
Pooling 3	Max Pooling	2×4×1		2×4×1	1×1×1	32	25×91×2×32
Convolution 4	Convolution	12×12×2		1×1×1	1×1×1	64	25×91×2×64
Pooling 4	Max Pooling	2×4×1		2×4×1	1×1×1	64	13×23×2×64
Convolution 5	Convolution	12×12×2		1×1×1	1×1×1	64	13×23×2×64
Pooling 5	Max Pooling	2×4×1		2×4×1	1×1×1	64	7×6×2×64
Convolution 6	Convolution	12×12×2		1×1×1	1×1×1	64	7×6×2×64
Pooling 6	Max Pooling	2×4×1		2×4×1	1×1×1	64	4×2×2×64
Fully Connected 1	Fully Connected					512	512
Fully Connected 2	Fully Connected					512	512
Output	Fully Connected					1	1

Table 1: CNN Architecture. Note the use of dilated convolutions in the upper layers in order to handle the very large input dimension. Pooling layers use equal stride and filter dimensions.

Hyperparameter	Value
Activation Function (Output Layer)	Linear
Activation Function (Other Layers)	Relu
Loss Function	Mean Absolute Error
Optimiser	Adadelata
learning rate	1
ρ	1
ϵ	0.0
Decay	0
Padding	Same
Weight Initialization	Glorot Normal
Bias Initialization	Zeros
Number of Epochs	170
Batch Size	18

Table 2: CNN Hyperparameters

where there are many sources of background signal, such as WiFi and Bluetooth. The hope was that the average pooling would have a smoothing effect on the data. Our preliminary testing showed that using average pooling in the first layer was able to lower the model's error, whereas replacing the max pooling operations lower down with average pooling had a detrimental effect.

The model was implemented using the Keras [10] framework, with TensorFlow [1] as its backend. The model was trained for 175 epochs, taking 4.5 days on an 1 Nvidia Titan Xp GPU.

3.3 Training Data Collection

All training data were collected on a single day. The collection was performed in rural farmland, near Lincoln, UK. Two drones were flown: a DJI Inspire 1² and a DJI Mavic Air³. The Inspire

was flown for three separate flights, each lasting the length of the battery charge (9 minutes and 20 seconds, 9 minutes and 53 seconds and 13 minutes and 40 seconds). The Mavic was flown for one such flight (19 minutes and 8 seconds). The second flight of the Mavic had to be cut short due to high wind conditions and so only lasted 8 minutes and 2 seconds. In order to gather more varied data, a Futaba controller was turned on and then driven at distance around the vicinity of the sensor. The operator of the drone controller had a smartphone (OnePlus 5T) on his person at all times. This smartphone logged the GPS coordinates of the controller every second, using the GPS Logger¹ app by BasicAirDelta. Using the GPS coordinates of the sensor, we were then able to calculate the bearing from the sensor to the controller, as described in Section 3.4.

During collection, we insured that the drone controllers were never less than 100 meters away from the sensor. This was to reduce the impact that inaccuracies in the GPS location would have on the calculation of the bearing between the sensor and the controller. Moreover, we attempted to keep the controller within 1km of sensor, as this is its expected range. While driving with the Futaba controller, some time was spent beyond this range limit. Through manual inspection of the data, we removed all time periods where the controller was out of range. During each drone flight, the operator was walking at a steady pace, in order to increase the variety of bearing labels provided to the system. Similarly, the operator flew the drone in a variety of patterns, spending some time hovering and other periods moving at a range of velocities.

If the sensor were deployed in urban areas, the frequency range in which it operates (2.4 - 2.4998GHz), would contain a lot of background signals from WiFi and Bluetooth. In order to recreate this, during certain periods of time, mobile WiFi hotspots were created close to the sensor. Other devices were connected to these hotspots

²www.dji.com/inspire-1

³www.dji.com/mavic-air?site=brandsite&from=landing_page

¹play.google.com/store/apps/details?id=eu.basicairdata.graziano.gpslogger&hl=en_GB

and downloaded items from the internet, in order to ensure that data was travelling over the network. Smartphones were also paired to bluetooth devices such as headphones.

3.4 Data Pre-Processing

The sensor produced data in the form of frames of dimension 3276×2 . Here, the first dimension is the number of frequency bins output by the sensor and the second represents the two values provided by each bin: power and AoA. Section 3.1 contains further details. The training data was normalised so that each value of each bin had zero mean and unit variance. The means and variances used for this normalisation were saved in order that future data used in testing could be normalised according to the same scheme. The radio signals of drones have certain unique temporal characteristics. For instance, the frequency of the control signal hops with a given period. See Section 2 for further details. In order for the model to learn these features, multiple frames had to be combined into windows. Preliminary testing found that windows composed of 200 frames provided a good trade-off between error and training time. The model was, therefore, fed windows of overall dimension $200 \times 3276 \times 2$.

Each window required an associated bearing label. During data collection, the GPS location of the sensor and the controller were recorded (see Section 3.3). From these values, for each controller GPS position record, the bearing between the sensor and the controller was calculated. As metadata, the hardware setup recorded the GPS time at which each frame was measured. Similarly, the GPS logging app used recorded the GPS time for each GPS measurement. Each window was then assigned the bearing value recorded closest in time to the middle, that is 100^{th} , frame. Given that the GPS coordinates of the controller were logged at a rate of 1Hz, the greatest possible mismatch in frame and GPS times was 0.5 seconds. The bearing values were normalised to between -1 and 1, corresponding to -180° and 180° , respectively.

3.5 Training Data Augmentation

We used two methods to augment the training data. The first of these was based on the construction of the time windows. Instead of starting the next time window from where the previous one finished (that is, 200 frames from the start of the previous time window), we started it some number of frames s from the start of the previous window, where $s < 200$. For this model, we decided to use a value of $s = 25$. This provided a roughly four times increase in the size of the training data. See Figure 6 for a diagram of this procedure.

The hardware system was constructed such that the calibration of the bearing outputs in each bin was not performed solely by the physical rotation of the sensor. Rather, this could also be performed in a software layer. The fact that the bearing calibration is not fully performed at a hardware level, meant that we could perform data augmentation by rotating our coordinate system. That is, we could shift the bearing readings in each bin, as well as the bearing labels, by a common angle α . In this work, we did two shifts to the data, by $\alpha = 120^\circ$ and $\alpha = 240^\circ$. This provided a three times increase in the size of the training data. After applying the data augmentation, we had a training data set of 15 606 examples.

s	d	$\beta = 90^\circ$	$\beta = 45^\circ$
100m	250m	28	26
	500m	74	79
300m	250m	29	13
	500m	52	40

Table 3: The implied mean absolute error (in meters) in position calculation based on a mean absolute error of 3.67° in bearing prediction by our model. See Figure 8 for a definition of symbols.

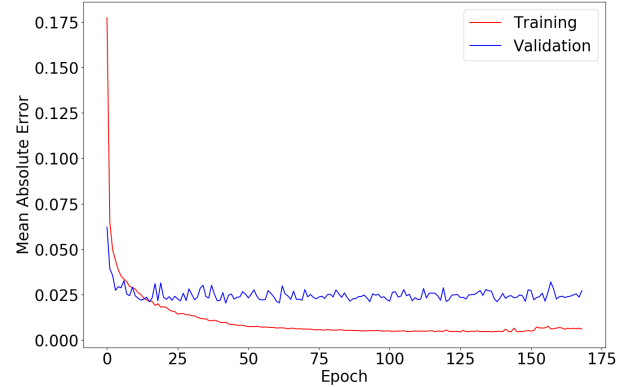


Figure 7: Training and validation error.

There has been recent work showing that data augmentation is an effective alternative to regularisation in many task domains [17]. Our preliminary testing demonstrated that to be the case in our particular domain. As such, no regularisation techniques were used. Our results in Section 4 show that our model was able to generalise very well, despite this lack of regularisation.

4 RESULTS

The model was trained on all the collected data, except for one flight of the Mavic drone, which was held out for model validation. Over 170 epochs, the epoch with the lowest validation error (measured by mean absolute error) had a validation error equivalent to 3.67° and a training error equivalent to 2.56° . A plot of training and validation errors is shown in Figure 7. It is worth calculating the localisation error implied by this bearing error. When calculating a position from two bearings, the error in the position calculation is given by equation (1) [22].

$$\varepsilon_p^2 = \frac{r_1 r_2}{\sin \theta} \varepsilon_{\alpha_1} \varepsilon_{\alpha_2} \quad (1)$$

Here, ε_p is the error in the position, r_1 and r_2 are the distances between the controller and the first and second sensors, ε_{α_1} and ε_{α_2} are the errors in the bearings from the first and second sensors, and θ is the angle between the two lines of bearing where they meet over the calculated position. Figure 8 contains a diagram of these variables. In order to calculate values for the positional error, we assume that the error in both sensors is equal for every prediction,

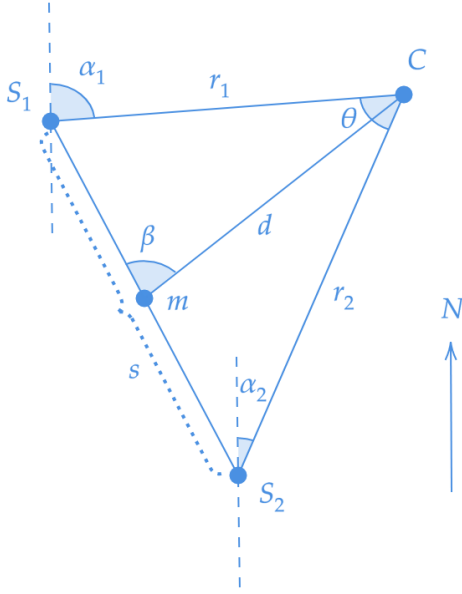


Figure 8: Symbol definitions for position error calculation. S_1 and S_2 are the locations of sensors 1 and 2. C is the location of the controller. m is the midpoint of the line connecting S_1 and S_2 .

that is $\varepsilon_{\alpha_1} = \varepsilon_{\alpha_2}$. We choose to calculate the errors for a number of distances d between the controller and the midpoint of the line joining S_1 and S_2 and a number of distances s between S_1 and S_2 . We also do so for two different value of the angle β , formed at the intersection of the line connecting the sensors and the line connecting the midpoint to the controller. The results of these calculations are shown in Table 3. The addition of extra sensors would reduce this error [22]. Moreover, it is worth noting that a large disadvantage of using two sensors is that the position error explodes as the angle β approaches $\pm 180^\circ$. However, it is the intention of our industrial partners to deploy these sensors in pairs, in order to reduce cost.

5 DISCUSSION AND CONCLUSIONS

We have presented a system which is capable of autonomously locating the controllers of UASs from their RF control signals. Although there has been substantial work on the detection of UASs, and the localisation of the aerial vehicles, as discussed in Section 1, to the best of the authors' knowledge, this is the first system capable of locating the controllers of UASs presented in the research literature. Moreover, it is capable of doing so almost instantaneously and with reasonable accuracy. As such, our system, or similar systems designed using the techniques put forward in this paper, could be effectively deployed in order to apprehend the operators of drones in dangerous and/or illegal situations. This would lead to a reduction in the harmful uses of drones, as mentioned in Section 1.

The implied mean absolute errors in position displayed in Table 3 show that our system is able to provide reasonably accurate locations, which could feasibly be used to apprehend an operator, at distances of up to 500m. Beyond this distance, the accuracy of the

location finding is fairly low. However, it does still provide much more information than only the knowledge that the controller must be within transmission range of the drone, which is a radius of a few kilometres.

The main strategy which we plan on following in order to reduce this error is to improve the labelling of the data. The GPS logging method described in 3.3 only had a refresh rate of 1Hz. This is not sufficient for the parts of the collection where the Futaba controller was held out of a moving car. Here, the car was travelling at upwards of 80km/h, due to surrounding traffic. Further strategies which we will be pursuing include the collection of more training data as well as the optimisation of our neural network architecture.

Given that the training data contained only three different controller types, it is possible that the model will not generalise well to other, unseen, controller types. Future work will focus on collecting data with a greater variety of controllers, as well as validating the model on unseen controller types. Future work will also aim to validate the system's error in a deployment situation.

A large focus of future work will also be the training of two further models, one for detecting the presence of UASs and the other for locating the drone itself. The three models will be combined into a greater system with a certain amount of higher-level control, defined programmatically. In the default mode of this system, data will be being fed into the detection model. When a drone is detected, the system will switch modes and start feeding data into the two localisation models.

REFERENCES

- [1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <https://www.tensorflow.org/> Software available from tensorflow.org.
- [2] Federal Aviation Authority. 2017. Reported UAS Sightings (July 2017-September 2017). www.faa.gov/uas/media/FY2017_Q4_De-identification_Redaction_11192017.xlsx
- [3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv preprint arXiv:1803.01271* (2018).
- [4] BBC. 2017. Ten sentenced for smuggling drugs into prisons by drones. www.bbc.co.uk/news/uk-42341416
- [5] BBC. 2018. Police drone finds man in ditch in Lincolnshire. www.bbc.co.uk/news/uk-england-lincolnshire-43188849
- [6] Andrea Bernardini, Federica Mangiardi, Emiliano Pallotti, and Licia Capodiferro. 2017. Drone detection by acoustic signature identification. *Electronic Imaging* 2017, 10 (2017), 60–64.
- [7] UK Airprox Board. 2017. *Analysis of Airprox in UK Airspace*. Technical Report.
- [8] CBS. 2015. Drones: a new favorite courier for smugglers. www.cbsnews.com/news/drones-smugglers-newest-way-prison/
- [9] François Chollet. 2018. *Deep learning with Python*. Manning Publications.
- [10] François Chollet et al. 2015. Keras. <https://github.com/keras-team/keras>.
- [11] Jerome T Connor, R Douglas Martin, and Les E Atlas. 1994. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks* 5, 2 (1994), 240–254.
- [12] Georg Dorrner. 1996. Neural Networks for Time Series Processing. *Neural Network World* 6 (1996), 447–468.
- [13] Lian Du, Chenqiang Gao, Qi Feng, Can Wang, and Jiang Liu. 2017. Small UAV Detection in Videos from a Single Moving Camera. In *CCF Chinese Conference on Computer Vision*. Springer, 187–197.
- [14] L Fuhrmann, O Biallawons, J Klare, R Panhuber, R Klenke, and J Ender. 2017. Micro-Doppler analysis and classification of UAVs at Ka band. In *Radar Symposium (IRS), 2017 18th International*. IEEE, 1–9.

- [15] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
- [16] pp Ruben Heersink. 2016. *An adaptive Bayesian approach towards a robust classifier for UAVs and birds*. Master's thesis. University of Twente.
- [17] Alex Hernández-García and Peter König. 2018. Data augmentation instead of explicit regularization. <https://openreview.net/forum?id=ByJWeR1AW>
- [18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [19] John R Hollenbeck, Daniel R Ilgen, Dale B Tuttle, and Douglas J Sego. 1995. Team performance on monitoring tasks: An examination of decision errors in contexts requiring sustained attention. *Journal of Applied Psychology* 80, 6 (1995), 685.
- [20] Business Insider. 2017. Exploring agricultural drones: The future of farming is precision agriculture, mapping, and spraying. uk.businessinsider.com/farming-drones-precision-agriculture-mapping-spraying-2017-8
- [21] Sungho Jeon, Jong-Woo Shin, Young-Jun Lee, Woong-Hee Kim, YoungHyoun Kwon, and Hae-Yong Yang. 2017. Empirical study of drone sound detection in real-life environment with deep neural networks. In *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 1858–1862.
- [22] Alonzo Kelly. 2003. Precision dilution in triangulation based mobile robot position estimation. In *Intelligent Autonomous Systems*, Vol. 8. 1046–1053.
- [23] Juhyun Kim, Cheonbok Park, Jinwoo Ahn, Youlim Ko, Junghyun Park, and John C Gallagher. 2017. Real-time UAV sound detection and analysis system. In *Sensors Applications Symposium (SAS), 2017 IEEE*. IEEE, 1–5.
- [24] Yongdae Kim. 2016. Security analysis of FHSS-type drone controller. In *Proc. Springer Int. Workshop on Information Security Applications*, Vol. 9503. 240.
- [25] Gihan J Mendis, Jin Wei, and Arjuna Madanayake. 2017. Deep learning cognitive radar for Micro UAS detection and classification. In *Cognitive Communications for Aerospace Applications Workshop (CCAA), 2017*. IEEE, 1–5.
- [26] Phuc Nguyen, Hoang Truong, Mahesh Ravindranathan, Anh Nguyen, Richard Han, and Tam Vu. 2018. Cost-Effective and Passive RF-Based Drone Presence Detection and Characterization. *GetMobile: Mobile Computing and Communications* 21, 4 (2018), 30–34.
- [27] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. 2017. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836* (2017).
- [28] Harvard Business Review. 2017. Drones Go To Work. hbr.org/cover-story/2017/05/drones-go-to-work
- [29] Muhammad Saqib, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein. 2017. A study on detecting drones using deep convolutional neural networks. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 1–5.
- [30] Waylon Dustin Scheller. 2017. Detecting drones using machine learning. (2017).
- [31] The Telegraph. 2016. Isil plotting to use drones for nuclear attack on West. www.telegraph.co.uk/news/2016/04/01/isil-plotting-to-use-drones-for-nuclear-attack-on-west/
- [32] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
- [33] Bas Vergouw, Huub Nagel, Geert Bondt, and Bart Custers. 2016. Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments. In *The Future of Drone Use*. Springer, 21–45.
- [34] Vox. 2017. Guess who has drones now? ISIS. www.vox.com/world/2017/5/30/15686240/drones-isis-iraq-syria
- [35] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. 2016. Deep Learning for Video Classification and Captioning. *arXiv preprint arXiv:1609.06782* (2016).
- [36] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.